JAVA PROGRAMMING BASICS

Module 2: Java Object-oriented Programming

Training program

1. Classes and Instances
2. The Methods
3. The Constructors
4. **Static Elements**
5. Initialization sections
6. Package
7. Inheritance and Polymorphism
8. Abstract classes and Interfaces
9. String processing
10. Wrapper classes for primitive types
11. Exceptions and Assertions
12. Nested classes
13. Enums
14. Generics
15. Collections
16. Method overload resolution
17. Multithreads
18. Core Java classes
19. Object Oriented Design
20. Functional Programming

Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

The static keyword 1/2

There are different uses of static keyword in java:

- static fields
- static methods
- static classes
- static blocks
- static import

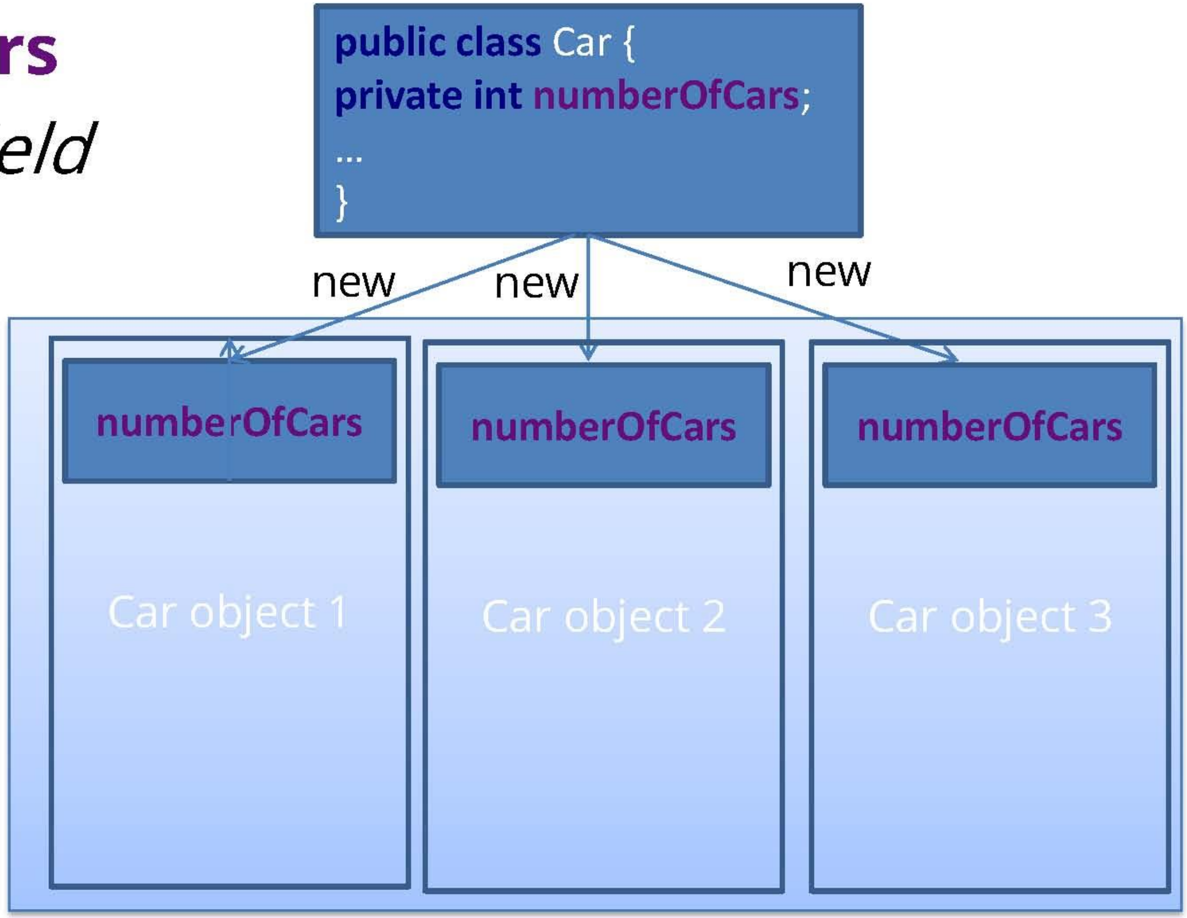
Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

The static fields 1/5

numberOfCars

***as** Instance Field*



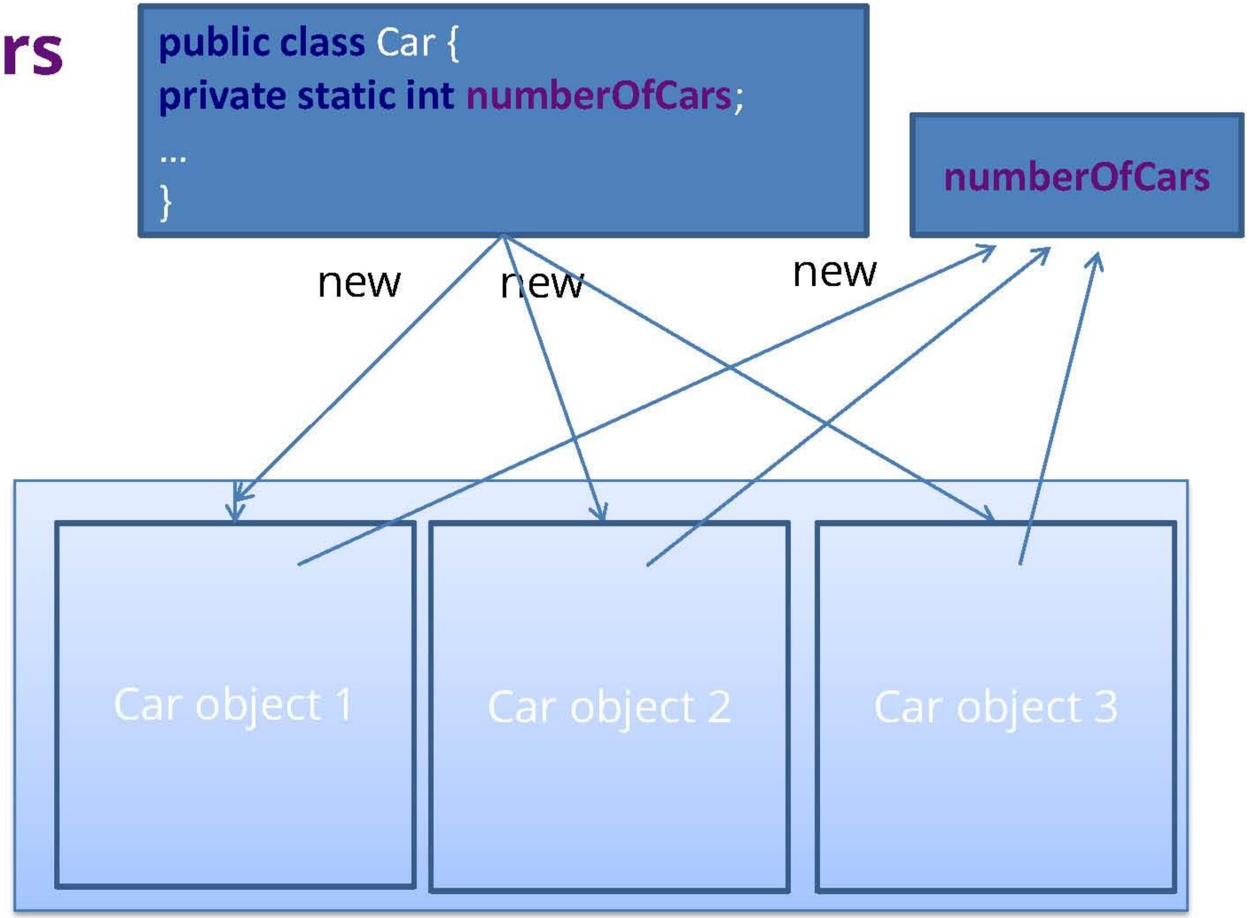
```
public class Car {
private int numberOfCars;
...
}
```

See BJStaticEduApp

The static fields 4/5

numberOfCars

as static Field



```
public class Car {
private static int numberOfCars;
…
}
```

Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

The static final fields 1/3

- The static modifier, in combination with the final modifier, is also used to define constants.

```
public class Car {
    //...
    public static final int NUM_OF_WHEELS = 4;
    //...
}
```

(compile-time constant - primitive or String)

The static final fields 2/3

- It is a compile-time error if a static final variable is not definitely assigned by a static initializer of the class in which it is declared.

```
public class Car {
    //...
    public static final int NUM_OF_WHEELS;
    //...
}
```

The static final fields 3/3

- The final modifier indicates that the value of this field cannot change.

```java
public class Car {
    public static final int NUM_OF_WHEELS = 4;
    //...
    public Car() {
        NUM_OF_WHEELS = 3;
    }
}
```

Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

The static methods 1/3

```
public class Car {
    private static int numOfCars;
    //...
    public Car() {
        numOfCars++;
    }
    //...
    public static int getNumOfCars() {
        return numOfCars;
    }
}
```

static method declaration

See BJStaticEduApp

The static methods 2/3

- Static methods should be invoked with the class name:

```
public class Main {
   public static void main(String[] arg) {
      Car myCar1 = new Car();
      Car myCar2 = new Car();
      Car myCar3 = new Car();
      int numOfCars = Car.getNumOfCars();
      System.out.println(numOfCars);
   }
}
```

static method invocation

The static methods 3/3

- You can also refer to static methods with an object reference but this is discouraged:

```
public class Main {
    public static void main(String[] arg) {
        Car myCar1 = new Car();
        Car myCar2 = new Car();
        Car myCar3 = new Car();
        int numOfCars = myCar1.getNumOfCars();
        System.out.println(numOfCars);
    }
}
```

static method Invocation (discouraged)

Module contents

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

The static methods restrictions 1/3

- Instance methods can access instance variables and instance methods directly.
- Instance methods can access class variables and class methods directly.
- Class methods can access class variables and class methods directly.
- Class methods can not access instance variables and instance methods directly.

See BJStaticEduApp

Using static fields and methods

- Static elements
  - The static keyword
  - The static fields
  - The static final fields
  - The static methods
  - The static methods restrictions
  - Using static fields and methods

Using static fields and methods

- Statics should be used when data is not instance dependent and for all existing instances of static member you want to apply same state.

- Static should be used for the methods where you have mostly utility methods in the class and we hardly need to maintain any object state to achieve the required action.

see objcreation